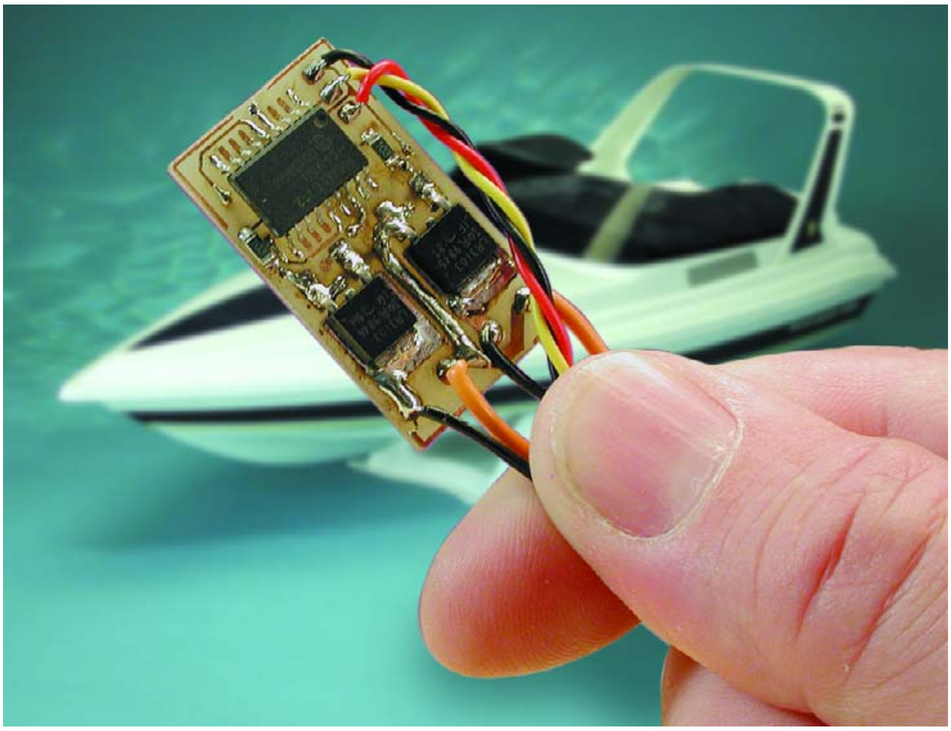
Add-on Switch Unit for R/C Models

Provides two switches at 30 A

Design by A. Pistorius

This switch unit is designed to drive two extra functions in a model via a switch-channel on the transmitter. A soft start function is provided on one of the channels for turning on electrical motors smoothly. The correct control signal and the supply are derived from the receiver in the model.



Radio controlled boats become a lot more interesting when, apart from the standard control functions, a few other functions can be controlled remotely. Most of you will be able to think of a few examples that would be useful. The simplest implementation is with a couple of micro switches that are actuated by a servo.

An alternative switch unit can be built using a monostable multivibrator (MMV) and a flip-flop. The MMV compares the incoming pulse width modulated signal with a reference pulse width (about 1.5 milliseconds). When the pulse is longer than 1.75 milliseconds, the flip-flop is triggered and the output becomes active. If this isn't the case then the MMV is reset. A second function can be similarly implemented by detecting pulses that are shorter than 1.25 milliseconds.

A suitable combination would be the 74LS123 (dual retriggerable MMV) and the 74LS73 (dual JK flip-flop), which would provide two extra functions. Standard CMOS components can also be used. Such a circuit can be made fairly small through the use of SMD components.

Switch units that are based on this principle are commercially available (such as the 'Duoswitch' made by Robbe), but they aren't cheap. Home construction is therefore recommended, using for example a popular servo IC, the ZN409 (or ZN419), which was used in the 'speed controller for model train' (*Elektor Electronics* December 1999). Unfortunately, this IC is no longer obtainable, so we'll have to think of a different way to tackle this problem.

For an alternative device that can be used in a switch unit you need look no further than a PIC microcontroller, such as the popular PIC16C712, made by Microchip. Usually the PIC16F84 is used in home construction circuits, but in this case we have specifically chosen the 16C712 because it includes PWM outputs. This happens to be the first time that this chip is used in an *Elektor*

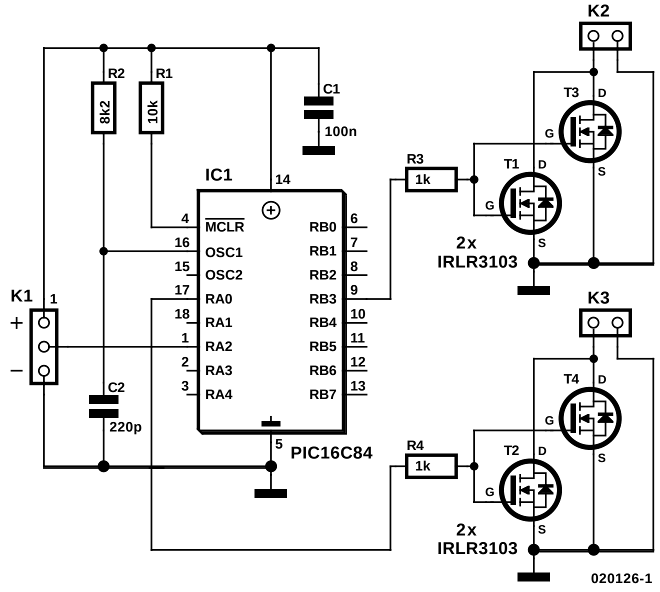


Figure 1. The circuit diagram of the switch unit.

Electronics circuit.

After this introduction the following description of the circuit won't come as a surprise: it has an input for the pulse width modulated signal and two outputs, which each drive an output stage with one or two FETs. An inexpensive RC clock generator completes the circuit. In this application there is no need to use a quartz crystal to obtain a very accurate clock frequency. As with many micro controller based circuits, the amount of soldering required is (sadly?) very limited.

The circuit diagram

The circuit diagram of the switch unit shown in **Figure 1** is conspicuous by its simplicity. The hart of the circuit is obviously IC1. This controller receives the servo signal from the receiver via K1. Outputs RB3 and RA0 each drive two FETs via a resistor. The FETs are connected to the outside world via K2 and K3.

The FETs used here are each capable of switching 20 A. This current is limited by the packaging, since the device itself is capable of switching a current up to 50 A! In order to provide a safety margin, it is recommended that the switched current is kept a bit below the stated 20 A. A value of 15 A per FET is more sensible. Since we're using two FETs per channel, each channel can deliver a continuous 30 A, and that's a substantial value! When a current of 15 A or less is required, either one of the FETs can be left out.

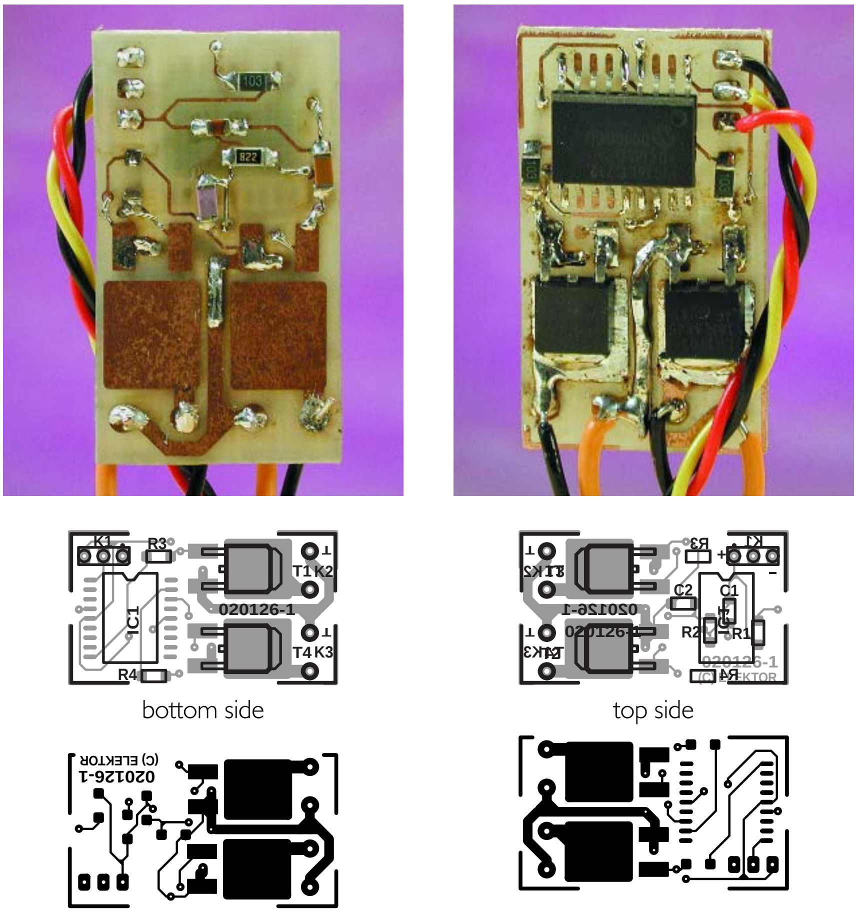
The RC network R2/C2 determines the operating frequency of the controller. The values shown here give a clock frequency of 400 kHz ±10%. The exact frequency isn't very important as the software still functions properly when the frequency deviates by a few percent.

R1 keeps the reset input of the controller at a logic High level and C1 prevents any interference produced by the controller from reaching the receiver. The circuit is powered via connector K1, which is connected to the receiver.

Figure 2 shows the printed circuit board for the switch unit. For modellers who have some experience with SMD components, the construction will be a piece of cake. The PIC16C712 can be obtained ready-programmed from the *Elektor Electronics* Readers' Services.

The software

The circuit diagram alone is not sufficient to explain the workings of the circuit. The PIC16C712 cannot function without a program. This also shows how powerful a microcontroller is compared to a dedicated IC such as the ZN409 that was mentioned earlier. Any restrictions of these types of IC can easily be overcome with a microcontroller. One of the disadvantages of the ZN409 was that there was no provision for a soft start output, without using extra components. With the use of a microcontroller these types of function can be implemented easily. This program uses one of the outputs (K2) as a soft start (and soft stop) output. This means that the switch-on doesn't happen abruptly, but is gradually increased using a PWM signal over a period of about 1 second. The switch-off also happens in a gradual manner. This method of switching is par-



bottom side

top side

Figure 2. Populating the PCB for this circuit is a routine task.

ticularly useful when an electrical motor is driven by the output. With a normal switch the motor will start abruptly. This obviously creates a large mechanical stress on the fixings of the motor. When the motor is provided with reduction gearing then this stress becomes much greater and could possibly lead to damage to the gearing and/or the fixings.

The software has been set up as a pulse width detector, using a counter that measures the period of a pulse at the input. The largest expected pulse width is about 2 milliseconds. If we use 200 instruction cycles for this then we can determine the pulse width to a high resolution. Furthermore, this counter value fits nicely in a single byte, keeping the program simple. This corresponds to a cycle time of 10 microseconds. The clock frequency required for this is 400 kHz. This frequency is obtained with a capacitor of 300 pF (in the prototype a capacitor was used with a tolerance of 5%) and a resistor of 6.2 kΩ. Of course it is possible to use different combinations as well, as long as the capacitor is greater than 20 pF and the resistor is between 3 kΩ and 100 kΩ (as recommended by Microchip). When an RC circuit is used for the clock generator the clock frequency (divided by 4) can be measured on pin 15 (OSC2/CLKOUT).

COMPONENTS LIST

Resistors:
R1 = 10kΩ
R2 = 8kΩ2
R3,R4 = 1kΩ

Capacitors:
C1=100nF
C2=220pF

Semiconductors:
IC1=PIC16C712-041/SO, programmed, order code **020126-41**
T1-T4= IRLR3101 (e.g., Farnell # 738-372)

Miscellaneous:
Servo cable
PCB, order code **020126-1** (see Readers Services page)
Disk, hex and source code files, order code **020126-11** or Free Download

And finally

Experimenting with microcontrollers completely changes the development cycle in modelling electronics. You need only modify the program to change the behaviour of the circuit. Those of you who have experience in programming PIC controllers in assembly language can download the source code from the *Elektor Electronics* website and modify it to your own needs. The source code includes sufficient comments to make its logic easily understood.

As can be seen from a recent article on modelling circuits (*Elektor Electronics*, February 2002) there is more than one road that leads to Rome. The circuit described here for the PIC 16C712 can again form the basis of various combined circuits (four inputs on PORTA and eight outputs on PORTB, with or without memory) and illustrates the simplicity and effectiveness of microcontroller circuits.

(020126)